

⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑫ 公開特許公報(A)

平2-138674

⑬ Int. Cl.⁵

識別記号

庁内整理番号

⑭ 公開 平成2年(1990)5月28日

G 06 F 15/70
15/22
15/62

3 3 0 G
3 2 5 K

7368-5B
7165-5B
8125-5B

審査請求 未請求 請求項の数 9 (全17頁)

⑮ 発明の名称 文書処理方法及び装置

⑯ 特 願 平1-214795

⑰ 出 願 平1(1989)8月23日

優先権主張 ⑱ 昭63(1988)8月24日 ⑲ 日本(JP) ⑳ 特願 昭63-209975

㉑ 発 明 者 河 内 谷 幸 子 神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

㉒ 発 明 者 武 田 晴 夫 神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内

㉓ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地

㉔ 代 理 人 弁理士 小川 勝男 外1名

明 細 書

1. 発明の名称

文書処理方法及び装置

2. 特許請求の範囲

1. 複数の文字と縦横の複数の罫線からなる表を含む文書を画像入力手段により文書画像に変換する第1ステップと、

上記文書画像を複数の罫線のみを含む第2の文書画像に変換する第2ステップと、

上記第2の文書画像に含まれる罫線を認識して、各罫線を定義する複数のベクトルデータを得る第3ステップと、

上記ベクトルデータに基づいて描かれた複数の罫線からなる表を表示する第4ステップとからなることを特徴とする文書処理方法。

2. 前記第2ステップが、文書画像に含まれる黒画素連結領域の外接矩形を求めるステップと、文書画像上で所定サイズ以下の外接矩形内に位置する黒画素を白画素に置換するステップとからなることを特徴とする第1請求項記載の文書

処理方法。

3. 前記第3ステップが、前記第2の文書画像から複数の横罫線を抽出して、各横罫線の始点と終点と、線幅とを示すベクトルデータを得るステップと、前記第2の文書画像から複数の縦罫線を抽出して、各縦罫線の始点と、終点と、線幅とを示すベクトルデータを得るステップとからなることを特徴とする第1請求項記載の文書処理方法。

4. 前記横罫線の抽出が、前記第2の文書画像を水平方向に順次に走査することにより検出されるそれぞれ1以上の黒画素からなる複数の線分のうち、所定の閾値よりも短い線分を除去することにより行なわれ、前記縦罫線の抽出が、前記第2の文書画像を垂直方向に順次に走査することにより検出される、それぞれ1以上の黒画素からなる複数の線分のうち、所定の閾値よりも短い線分を除外することにより行なわれることを特徴とする第3請求項記載の文書処理方法。

特開平2-138674 (2)

5. 前記各横罫線のベクトルデータが、前記水平方向の走査により検出される所定閾値より長い複数の線分のうち、互いに接する位置関係にあるものを統合して、線幅または線長が大きくなった1つの線分にし、この統合された線分の始点、終点、線幅を検出することにより求められ、前記各縦罫線のベクトルデータが、前記垂直方向の走査により検出される所定閾値より長い複数の線分のうち、互いに接する位置関係にあるものを統合して、線幅または線長が大きくなった1つの線分にし、この統合された線分の始点、終点、線幅を検出することにより求められることを特徴とする第4請求項記載の文書処理方法。
6. 第3請求項記載の文書処理方法において、更に、前記第3ステップで得られた複数のベクトルデータのうち、縦罫線を示すベクトルデータと、横罫線を示すベクトルデータとを照合し、始点または終点の値を選択的に補正するステップを有することを特徴とする文書処理方法。
7. 第1請求項記載の文書処理方法において、更

に、第1ステップで得られた文書画像の傾きを検出し、傾きの補正するステップを有し、前記第2ステップが上記傾き補正された文書画像に対して行なわれるようにしたことを特徴とする文書処理方法。

8. 第1請求項記載の文書処理方法において、更に、前記第3ステップで得られた各罫線のベクトルデータに基づいて、それぞれ前記表を構成する上下、左右が罫線で区切られた複数のセクションを検出し、各セクションを定義するデータを得るステップを有することを特徴とする文書処理方法。

9. 表と文字とを含む文書画像を入力するための画像入力装置と、

文書画像を表示するための画面をもつ表示手段と、

上記画像入力手段から入力された複数画面からなる文書画像を記憶するための第1のメモリ手段と、

傾きを補正された文書画像を記憶するための

第2のメモリ手段と、

罫線を定義するベクトルデータを記憶するための第3のメモリ手段と、

オペレータによって操作される指令と文字データをを入力するための手段と、

上記入力手段からの指令にตอบสนองして動作するデータ処理手段と、

からなり、上記データ処理手段が、上記第1のメモリ手段に記憶された文書画像の傾きを検出し、傾き補正された文書画像を上記第2のメモリ手段に書込む第1手段と、文字を構成している黒画素を白画素に置換することにより、上記第2のメモリ手段に記憶されている文書画像を、複数の罫線が残されている第2の文書画像に変換するための第2手段と、上記第2の文書画像から、横罫線のグループと縦罫線のグループとを別々に検出し、各罫線の位置とサイズを定義する複数のベクトルデータを上記第3のメモリ手段に書込む第3手段と、上記ベクトルデータに基づいて描かれた複数の罫線からなる表を、

上記表示手段に出力するための手段とを備えたことを特徴とする文書処理装置。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は文書処理方法および装置に関し、更に詳しくは、複数の罫線からなる表と文字とを含む文書画像から罫線を認識して、表作成用のデータを得るようにした文書データ処理方法および装置に関する。

〔従来の技術〕

ワードプロセッサやワークステーションなどの文書作成装置において、表を含む文書を作成しようとする、通常は装置を罫線モードに設定し、罫線の始点と終点をそれぞれカーソルにより指定することにより罫線を引き、この操作を縦、横の複数の罫線に繰り返している。また、装置によっては、カーソルで始点を指定し、該始点から右方向、または下方向に延びる罫線を引き、次に終点をカーソルで指定し、終点より右、または下に延びた不要な罫線部分を消去することにより、所望

特開平2-138674 (3)

長さの罫線を描いている。このようにして描かれた表の枠外や、表内の所定の欄(セクション)に文字を記入する場合には、罫線モードを解消して装置を文字入力モードに戻し、文字キーとカーソル・キーの操作により文字を入力する。

表を描くための面倒なキー操作をなくするための1つの方法として、例えば特開昭62-138988号公報において、白紙上にフリーハンドで比較的きれいに描いた表を、画像入力装置により読取り、表認識のための特殊な知識を蓄えている知識ベースを利用して、上記入力された表画像を認識し、表の枠を自動的に生成するアイデアが提案されている。

【発明が解決しようとする課題】

然るに、入力画像を利用して表を生成する上記従来の装置は、入力画像が複数の罫線からなる表のみを含むことを前提として表認識がなされており、例えば、既存の伝票や文書の如く、文書と表とが混在する用紙を原稿として用いることはできない。

線データとを自動的に生成する文書処理方法、および装置を提供することにある。

【課題を解決するための手段】

上記目的を達成するために、本発明による文書処理方法では、複数の文字と、縦横の複数の罫線からなる表を含む文書を、画像入力手段により文書画像に変換する第1ステップと、上記文書画像を、複数の罫線のみを含む第2の文書画像に変換する第2ステップと、上記第2の文書画像に含まれる罫線を認識して、各罫線を定義する複数のベクトルデータを得る第3ステップと、上記ベクトルデータに基づいて描かれた複数の罫線からなる表を表示する第4ステップとからなることを特徴とする。

更に詳述すると、上記第2ステップは、例えば、文書画像に含まれる黒画素連結領域の外接矩形を求めるステップと、文書画像上で所定サイズ以下の外接矩形内に位置する黒画素を白画素に置換するステップとからなる。また、上記第3ステップは、上記第2の文書画像から複数の横罫線を抽出

本発明の目的は、複数の罫線からなる表と、表の内、あるいは外に記入された複数の文字とを含む文書を原稿として用い、画像入力装置により読込まれた文書画像から罫線を認識して、自動的に罫線を描くようにした文書処理方法、および装置を提供することにある。

本発明の他の目的は、罫線の一部が劣化したり、縦横の罫線が正確に交っていない文書画像を処理して、鮮明で正しい交点をもつ罫線からなる表を生成できるようにした文書の処理方法、および装置を提供することにある。

本発明の更に他の目的は、複数の罫線と文字列とを含む文書画像から罫線を認識し、ワードプロセッサ機能によりオペレータが罫線を修正したり追加したりできる形の罫線データを自動的に生成する文書処理方法、及び装置を提供することにある。

本発明の更に他の目的は、複数の罫線と文字列とを含む文書画像から罫線と、これらの罫線により囲まれた領域とを認識し、罫線定義データと個定

して、各横罫線の始点と、終点と、線幅とを示すベクトルデータを得るステップと、上記第2の文書画像から複数の縦罫線を抽出して、各縦罫線の始点と、終点と、線幅とを示すベクトルデータを得るステップとからなる。

本発明による文書処理装置は、表と文字とを含む文書画像を入力するための画像入力装置と、文書画像を表示するための画面をもつ表示手段と、上記画像入力手段から入力された複数の画素からなる文書画像を記憶するための第1のメモリ手段と、傾きを補正された文書画像を記憶するための第2のメモリ手段と、罫線を定義するベクトルデータを記憶するための第3のメモリ手段と、~~罫線を定義するベクトルデータを記憶するための第2のメモリ手段と~~、オペレータによって操作される指令と文字データを入力するための手段と、上記入力手段からの指令にตอบสนองして動作するデータ処理手段とからなり、上記データ処理手段が、更に、上記第1のメモリ手段に記憶された文書画像の傾きを検出し、傾き補正された文書画像を上記第2の

特開平2-138674 (4)

メモリ手段に書込む第1手段と、文字を構成している黒画素を白画素に置換することにより、上記第2のメモリ手段に記憶されている文書画像を、複数の罫線が残されている第2の文書画像に変換するための第2手段と、上記第2の文書画像から、横罫線のグループと縦罫線のグループとを別々に検出し、各罫線の位置とサイズを定義する複数のベクトルデータを上記第3のメモリ手段に書込む第3手段と、上記ベクトルデータに基づいて描かれた複数の罫線からなる表を上記表示手段に出力するための手段とを備えることを特徴とする。

【作用】

本発明において、前記横罫線の抽出は、例えば第2の文書画像を水平方向に順次に走査することにより検出されるそれぞれ1以上の黒画素からなる複数の線分のうち、所定の閾値よりも短い線分を除外することにより行なわれ、前記縦罫線の抽出は、第2の文書画像を垂直方向に順次に走査することにより検出されるそれぞれ1以上の黒画素からなる複数の線分のうち、所定の閾値よりも短

かい線分を除外することにより行なわれる。また、前記各横罫線のベクトルデータは、水平方向の走査により検出される所定閾値より長い複数の線分のうち、互いに接する位置関係にあるものを統合して、線幅あるいは線長が大きくなった1つの線分にし、この統合された線分の始点、終点、線幅を検出することにより求められる。同様に、前記各縦罫線のベクトルデータは、垂直方向の走査により検出される所定閾値より長い複数の線分のうち、互いに接する位置関係にあるものを統合して、線幅または線長が大きくなった1つの線分にし、この統合された線分の始点、終点、線幅を検出することにより求められる。

上述したベクトルデータに基づいて罫線を描くことにより、本発明によれば、表示画面およびプリンタに劣化のない罫線からなる鮮明な表を出力させることができる。

尚、表示された表内の任意のセクションを、オペレータがカーソルで指示し、このセクションに文字データを入力できるようにするためには、カ

ーソルがどのセクションに位置しているかを判断するためのセクション定義データが必要となる。本発明によれば、表における各セクションの定義データを、前記第3ステップで得られた各罫線のベクトルデータ、または、罫線のはみ出し部分を補正したベクトルデータに基づいて、例えば最上位の横線を基準線として、これと交わる縦線を順次見つけ、次にこれらの縦線と交わる横線を順次見つける形で、上記基準線を上辺とする第1行目の複数のセクションの位置データを算出し、次に第2番目以降の横線を順次に基準線として同様の動作を繰り返すことにより求めることができる。

【実施例】

以下、本発明の1実施例を図面を参照して説明する。

第1図は、本発明による文書処理方法を実施するシステムの全体構成の1例を示す。このシステムは、オペレータがコマンドやデータを入力するためのキーボード11と、帳票などの文書画像を入力するためのイメージスキャナ12と、入力文

書あるいは、このシステムにより認識処理された文書データを表示するための表示装置13、プリンタ14、プログラムメモリ16にストアされたプログラムに従ってデータ処理動作を行なうデータプロセッサ15、文書認識の結果得られたベクトルデータ（線分データ、幅データ）や文字データをストアするための文書データファイル17、画像入力装置12から入力された画像データを1時的にストアするための入力バッファメモリ18、上記入力バッファメモリ18にストアされた画像データを処理して得られる補正画像データをストアするためのメモリ19、および本発明による文書認識処理に必要な各種のテーブルとワークデータをストアするためのメモリ20とからなる。メモリ20は、文書画像から検出した黒画素連結領域の外接矩形枠を示すデータをストアするためのテーブル領域21、文書に含まれる罫線を定義するデータをストアするためのテーブル領域22、文書に含まれる表を構成する欄を定義するデータをストアするためのテーブル領域23、歪んだ罫

特開平2-138674 (5)

線を補正する処理に用いられる横仮線データテーブル領域24および縦仮線データテーブル25、横罫線と縦罫線との識別のために用いられる線長と傾度テーブル領域26、横線検出の閾値をストアするための領域27、縦線検出の閾値をストアするための領域28、外接矩形枠検出に用いるラベルデータをストアするための領域29、およびワークエリア30からなる。

第2A図～第2G図は、本発明による文書認識処理の手順を模式的に示した図である。

第2A図は、処理対象となる原帳票用紙200を示し、この帳票は罫線210と、見出し文字列211と、欄内に記入された文字データ212とを含む。第2B図は、画像入力装置12により入力画面バッファ18に読み込まれた上記帳票の入力画像データ201を示す。入力画像データ201は、オペレータによる入力装置12への紙面セットの不正確さに起因して、水平（あるいは垂直）方向に対して傾いた状態となる。第2C図は、入力画像201に含まれる直線（罫線の1部）

210を検出し、傾きを補正して得られる画像データ202を示す。この画像データ202は、原帳票用紙（原稿）200に存在する印刷のよごれ、および画像入力装置12における変換処理時に生ずるノイズなどにより、罫線の1部に突起213や欠落214、などの欠陥が生じている。

第2D図は、入力画像202に含まれる黒画素の連結領域を検出し、それぞれの黒画素連結領域の外接矩形枠を求め、外接矩形枠のサイズから文字枠を識別し、文字情報211、212を除去することにより得られた罫線情報210のみを含む画像データ203を示す。

第2E図は、画像データ203から横線と縦線を検出した後、各線分のベクトルデータを求め、これらのベクトルデータに基づいて描画したグラフィックデータ画像204を示す。横線の検出は、例えば、画像データ203を横方向に順次に走査することにより同一走査線上で連続する複数画素からなる仮線を求め、連続画素数が所定値以下のものは縦罫線要素とみなして除外し、残された複

数の仮線を縦罫線構成要素と判断する。複数の互いに隣接する仮線を1つの実線に統合し、その始点と終点とから各横罫線のベクトルデータを求める。これと同様に、縦線のベクトルデータは、画像データ203を縦方向に走査し、同一走査線上で連続する所定個数以上の画素からなる縦方向の仮線を求め、隣接仮線を統合することにより求める。これらの画像処理により、罫線は図215'に示す如く、欠落の補正された直線となるが、213'で示すように、縦線と横線の交点が部分的に突出したり、未接続になってりした不完全な状態である。

第2F図は、上記グラフィックデータ画像204に対して補正処理を施し、突起がなく、縦線と横線が正しく交わり、且つ、罫線が所定の線幅に統一された、修正されたベクトルデータにより描かれたグラフィックデータ画像205を示す。

第2G図は、縦、横の罫線により形成される罫（例えば斜線で示した領域216は1つの罫である）を認識し、オペレータがカーソルで指示した

位置がどの罫に相当するかを判断するのに必要な測定値データを求めた状態を示す。

第3図は、上述した文書認識処理を行なうための処理プログラムのフローチャートを示す。

このプログラムでは、まず、画像入力装置12により入力帳票200の画像情報を入力画像バッファメモリ18に読み込む（ステップ301）。次に、罫線210を利用して、第4図と第5図を参照して後述する入力画像201の傾きを検出処理を行ない（ステップ302）、第6図で詳述するように傾き補正された画像データ202を画像データ領域19にストアする（ステップ303）。傾き補正された画像データ202は表示装置13に出力され（ステップ304）、オペレータ（ユーザ）は、上記表示画像を視覚的にチェックする。オペレータは、表示画面をみて、帳票画像を再入力するか否かを判断し、もし、表示された傾き補正画像に問題がなければOKを示すコマンドをキーボード11から入力する。OKの入力があった場合（ステップ305）、画像データ202に含

特開平2-138674 (6)

まれる黒画素連続領域の外接矩形を求めて、文字に外接する枠を認識し(ステップ306)、次に文字情報と罫線(直線)情報とを分離して、画像データ202から文字を消去した第2D図に示す画像203を得る(ステップ307)。次に、第9A図～第20図で詳述するように、画像データ203に含まれる罫線から縦線と横線を識別し、各線分のベクトルデータを得て(ステップ308)、これらのベクトルデータに基づいて描画される第2E図のグラフィックデータ画像204を表示画面に表示する(ステップ309)。更に、第21図で詳述するように、画像204に含まれる罫線の各交点に対する補正処理と、線幅の統一処理を行ない、第2F図に示す補正されたグラフィックデータ205画像を得(ステップ310)、このグラフィックデータ画像に対して、第22図～第26図で説明する罫線の認識処理を行なう(ステップ312)。尚、グラフィックデータ画像204または205に対しては、オペレータは、ステップ313で示す如く、キーボードから罫線の補正

操作を加えることができ、この場合、オペレータの操作コマンドに応じて線分のベクトルデータが修正、消去あるいは追加される。

以下、上記フローチャートの主たるステップの詳細について説明する。

ステップ302で行なう傾き検出では、入力画像201に含まれる罫線(直線)の1つを検出し、この直線の傾きを求める。画像に含まれる直線の検出は、公知の種々の方法で行なうことができるが、ここでは公知のHough変換のアルゴリズムを適用する。

第4図は、左辺に相当する罫線を検出するためのプログラム・フローチャート、第5図は、その説明図を示す。第5図で、 $y = x \sin \theta + y \cos \theta$ が求める直線の式であり、 y は原点Oからの距離、 θ は直線の傾きを表わす。Hough変換方式の特徴は、画像中の、例えば文字などの他の情報に影響されることなく、直線を検出できることにある。第4図のフローチャートにおいて、ステップ401、402はパラメータの初期化であり、ス

テップ403～406で直線上の1つの点の候補、例えば黒画素 P_1 を探索し、次にステップ407～411で、この点 P_1 を通る全ての直線として、 $y = x \sin \theta + y \cos \theta$ における y と θ との組を求め、更に、ステップ403～412を繰り返すことにより、傾度分布 $f(y, \theta)$ を求める。ここで $f(y, \theta)$ は、直線 $y = x \sin \theta + y \cos \theta$ 上に位置する候補点の数を示す。例えば、第5図の例では、点 $P_1 \sim P_4$ を通る直線 (y, θ) に対しては $f(y, \theta)$ の値は“4”となるが、これ以外の直線 (y, θ) に対しては、 $f(y, \theta)$ の値は0～2の小さい値となる。従って、ステップ413で $f(y, \theta)$ の値が最大となる y, θ を求めることにより、最も多くの候補点を通る直線のパラメータが得られる。

第3図のステップ303で行なう傾きの補正は、例えば、第6図に示す如く、補正画像202上の点Aを入力画像201上の点aの座標に変換し、点Aの画素濃度として、例えば、点aに最も近い点a'の画素濃度を与える操作を、補正画像上の

全ての点(画素)について繰り返すことにより行なう。点Aに対応する点aの座標値は、ステップ302で求めた直線の傾き角度 θ を用いて計算できる。尚、点aの濃度値は、点aを囲む4つの画素の濃度値と、点aとこれらの画素の距離とに応じて計算してもよい。

第3図のステップ306に示した文字外接矩形枠の認識は次のようにして行なう。

第7A図は、罫線210と、文字212A～212Cを含む画像202の拡大図である。本発明で言う文字外接矩形枠(文字枠)は、220A～220Cの如く、文字を構成する互いに連結した複数の黒画素に外接する矩形を意味し、これらの文字枠の始点と終点は、それぞれ左上画素 P_s と右下画素 P_e で示す。各文字外接矩形枠は、例えば、電子通信学会編、コロナ社発行、「パターン情報処理」(長尾真著)、P.84-85に記載された方法で検出できる。上記方法によれば、画像202を上側の行から順に、各行では左から右に走査し、各行で、連続する黒画素が互いに同

特開平2-138674 (7)

一のラベル（識別符号）をもつように順次にラベル付けをしていく。

例えば、第7B図に示す如く、画像202を矢印方向に順に走査し、最初に検出された黒画素の位置にはラベルAを与え、これに連続する黒画素位置には同じラベルAを与えていく。黒画素が途切れた後、同一の行内で別の黒画素が現われたら、これには別ラベルBを与え、これに続く黒画素の位置に同一ラベルBを与えていく。これらの黒画素の位置とラベルとの関係はメモリ領域29に記憶しておく。次の行からは、黒画素が検出される毎に、その近傍に既にラベル付けされた点があるか否かを判定し、もし有れば上記ラベルと同一のラベルを与え、なければ新たなラベルを与えていく。このようにラベル付けを行なうと、第7B図に示す如く、同一ラベルをもつ互いに連続する黒画素の集合 $S_A \sim S_D$ が形成される。ここで、例えば S_C と S_D の如く、異なるラベルを持つ複数の集合が連続している場合、ラベルを統一する必要がある。そこで、画像202全体の走査を終えた時、

メモリ領域29をチェックし、他の集合 S_C と連結した集合(S_D)について、ラベルの書き換え($D \rightarrow C$)を行なう。そして、同一ラベルを持つ各画素集合の外接矩形220を求め、それぞれの始点 P_s と終点 P_e の座標を外接矩形枠テーブル21に、例えば第8図に示すように記憶する。

第8図において、21Aは矩形番号、21Bは始点座標、21Cは終点座標である。外接矩形テーブル21には、文字を構成する黒画素の外接矩形データと、罫線を構成する黒画素の外接矩形データとが記憶される。そこで、始点座標21Bと終点座標21Cとから、各矩形のサイズを求め、所定のサイズより小さい矩形を文字外接矩形と判断する。

第3図のステップ307は、画像202に含まれる黒画素のうち、上記文字外接矩形内に位置するものを白画素に変換する処理である。これによって、罫線210だけを残した画像203が得られる。

次に、第3図の罫線認識ステップ308の詳細

について、第9A図～第20図を参照して説明する。

第9A図は、画像入力装置12を通る前の原画像200上の原罫線210の拡大図を示す。原罫線210は鮮明であり、線幅に乱れはない。

第9B図は、画像入力装置12により読み込まれ、文字の除去処理を終えた後の画像203に含まれる罫線210の拡大図である。この罫線210はノイズ等により若干劣化しており、連続すべき黒画素の1部が欠けた状態となっている。

罫線認識処理308は、第9B図に示す劣化した罫線画像から、第9A図に示す劣化のない罫線を描画するためのデータを得るためのものであり、本発明では、第10図に示す如く、罫線210の中の横線（例えば、 $LH1, LH2 \dots$ ）を認識するルーチン500と、縦線（例えば、 $LV1, LV2 \dots$ ）を認識するルーチン510と、罫線テーブル（実線ソフト）22を作成するルーチン520とからなっている。

横線認識ルーチン500の詳細を第11図に示

す。このルーチンでは、まず、画像203に含まれる仮線を抽出し、仮線テーブル24を作成する（ステップ501）。ここで言う“仮線”は、第9B図に $L1, L2, \dots$ で示すように、横方向の走査線上で連続する複数の黒画素からなる線幅が1画素の線分を意味し、1本の罫線は、その線幅に応じた複数本の仮線から形成される。ステップ501では、画像203を水平方向に走査しながら、各走査線上に存在する複数の連続黒画素からなる場合の始点（例えば PA ）と終点（例えば PE ）を検出し、これら2点の座標と、線幅（＝1画素）、線長（ $PA \sim PE$ の距離）の値を第12図に示すテーブル24に順次に登録する。画像203の全体について走査が終了すると横罫線を構成する仮線だけでなく、縦罫線を構成する仮線も含めて、全ての仮線データ24B～24Lが仮線の検出順の線番号24を伴ってテーブル24に登録される。

次に、横罫線を構成する仮線（横仮線）と縦罫線を構成する仮線（縦仮線）とを見分けるために、

特開平2-138674 (8)

仮線テーブル24から線長データ24Lを読み出し、線長毎の仮線の頻度（出現した本数）を表わす頻度リスト26を作成する（ステップ502）。第13図は頻度リスト26の1例を示す。一般に、縦、横罫線を含む画像で、横方向に長さを測って仮線の頻度分布をとると、縦罫線は短い線長をもつ多数の線分に分割されているため、縦罫線を構成している短い仮線が高い頻度をもつことになる。例えば、各縦罫線が1画素分の線幅をもつ場合は、線長が1（画素）の仮線の頻度が最大となり、縦罫線の線幅が2画素ならば、線長が2の仮線の頻度が最大となる。従って、最大頻度をもつ線長 ℓ を縦罫線の線幅とみなし、上記線長 ℓ 、あるいは上記線長 ℓ に所定の係数 k （例えば $k=2$ ）を掛けた値を、横仮線・縦仮線を識別するための閾値 TH としてメモリ領域27に記憶しておく。

仮線テーブル24は、線長24Lが上記閾値 TH より大きい仮線データ、すなわち横仮線データのみを残し、縦仮線データをテーブルから消去した形に再編換してもよいが、この実施例では、

示す定義に従って、基準仮線の始点、終点、線幅、線長さの値を更新し（仮線と基準仮線との統合）、統合された仮線のデータを仮線リスト24から消去することにある。

第14図で、 k は基準仮線を示すパラメータ、 ℓ は照合対象となる仮線を示すパラメータであり、ステップ1101と1113により1つの k についてのループが形成され、ステップ1103と1111とで1つの ℓ についてのループが形成されている。この例では、統合された仮線は、リストから消去する代りに、始点の x 座標に「-1」を代入している。また、ステップ1102、1104で基準仮線の長さ $Length(k)$ 、照合仮線の長さ $Length(\ell)$ を閾値 TH と比較し、これが TH より小さい場合は処理対象からはずすようにしている。事前に仮線テーブルから縦仮線のデータが消去されている場合は、閾値 TH との比較動作を省略すればよい。

このルーチンでは、ステップ1101で k に初期値を設定し、ステップ1102で k 番目の仮線

縦仮線データを仮線テーブルに残したまま、次の仮線統合処理ステップ504を実行する。

仮線テーブル24上では、各横罫線 $LH1$ 、 $LH2$ ……が、それぞれ複数の仮線データとして記憶されている。そこで、ステップ504では、隣接関係にある複数の横仮線を統合し、複数画素分の線幅をもつ1本の横罫線を表わす実線データに変換する。

第14図は、ステップ504で行なう横仮線統合処理の詳細を示すフローチャートである。ここで、仮線テーブル24における k 番目の仮線の始点座標24Bの値を $[x1(k), y1(k)]$ 、終点座標24Eの値を $[x2(k), y2(k)]$ 、線幅24Wの値を $Width(k)$ 、線長24Lの値を $Length(k)$ でそれぞれ表わすことにする。

第14図のフローチャートの概要は、仮線リスト24の中で1つの横仮線を基準仮線に選び、これ以外の全ての横仮線について順次に上記基準仮線と接するか否かをチェックし、もし、基準仮線と接するか重なる場合は、第15図と第16図に

データが基準仮線として有効か否かを判定する。
 k 番目のデータが実質的に消去されたもの、~~あるいは k 番目のデータが実質的に消去されたもの、~~

あるいは縦仮線に該当する場合は、ステップ1112に分岐し、 k の値を1だけインクリメントする。インクリメントされた k の値が、仮線テーブル24にストアしてある。データの個数 k_{max} を超えた場合、このルーチンを終了する（ステップ1113）。もし、そうでなければステップ1102に戻る。ステップ1102で、 k 番目の仮線データが基準仮線として有効な場合、ステップ1103で、照合仮線を示すパラメータ ℓ に初期値を設定し、ステップ1104、1105で ℓ 番目の仮線データが有効か否かを判定する。もし、有効データでなければ、ステップ1110で ℓ の値を1だけインクリメントし、これが仮線リストのデータ個数 $\ell_{max}(=k_{max})$ を超えていなければ、ステップ1104に戻り、そうでなければ、ステップ1112に進む。 ℓ 番目のデータが有効な場合、ステップ1106で基準

特開平2-138674 (9)

仮線と照合仮線との位置関係をチェックし、もしこれら2本の仮線が重なる場合、あるいは互いに接する場合、基準仮線の始点、終点の座標変更(ステップ1107)と、線幅の変更(ステップ1108)を行ない、照合仮線データを消去することを意味する始点X座標の「-1」への書き換えを行なう。

第15図は、基準仮線(k番目の仮線)と照合仮線(l番目の仮線)とが互いに接するか、あるいは互いに重なる場合の、基準仮線の始点x座標 $x_1(k)$ と終点x座標 $x_2(k)$ の変更の定義を示す。照合仮線の始点または終点が、基準仮線の外側に位置する場合は、基準仮線の始点または終点のx座標が照合仮線の始点または終点のx座標に置き換えられる。基準仮線と照合仮線とが統合されると、基準仮線は、始点または終点のx座標が変更されると同時に、線幅も変更されて、次第に太くなって行く。

例えば第17図の画像において、L1が基準仮線に選ばれた場合、仮線L2は重ならないため、

った場合の線幅の変更の定義を示す。線幅 $W(k)$ と $W(l)$ もつ2つの線分の重なり方には、図示された3通りがあり、照合仮線が基準仮線に完全に隠れる第3ケースを除いて、y座標で表わすと

$$\text{Width}(k) = y_2(l) - y_1(k) + 1$$

の関係にある。

上述のデータ処理により、第12図に示した横仮線テーブル24の内容は、例えば第18図の如く変更される。

第10図の擬線認識ルーチン510は、上述した横線検出ルーチンにおけるxとyとの関係を入れ換え、画像203を縦方向に走査して得られる仮線データを縦仮線テーブル25に記憶し、これらの仮線データに対して第14図と同様の処理を施せばよい。この場合、縦仮線と横仮線とを識別するための閾値 TH' はメモリ領域28に記憶しておく。

第10図の実線リスト作成ステップ520では、

そのまま仮線リストに残る。仮線L3は基準仮線に重なるため、これが基準仮線に統合され、基準仮線L1の終点座標がステップ1107で P_{A_1} から P_{B_1} に変更される。また、基準仮線L1の線幅がステップ1108で「2」に変更され、仮線L3のデータがステップ1109で消去される。仮線L4は、線幅が拡張された基準仮線に重ならないため、この時点では仮線リストに残されたままである。

基準仮線L1と他の全ての仮線との照合動作が終了すると、次に、仮線L2が基準仮線に選ばれる。この基準仮線L2は、既にL3と統合済みの仮線L1と照合され、両者が互いに重なっているため、L2の始点座標 P_{A_2} が P_{A_1} に変更される。また、L2の線幅が「2」に変更され、仮線L2は仮線リストから消去される。次に、仮線L4が照合仮線になり、これも基準仮線L2に統合され、結果的に仮線L1～L4が統合されて1本の実線となる。

第16図は、基準仮線kと照合仮線lとが重なり

横仮線テーブル24に記憶されている統合された横仮線データと、縦仮線テーブル25に記憶されている統合された縦仮線データとを擬線テーブル(実線リスト)22に書き込む。上記処理は、横仮線テーブル24に記憶されている横仮線データを順次に読み出し、始点のx座標が「-1」のもの、および線長 $24W$ が閾値 TH より大きいものだけを選択して、第19図の如く擬線テーブル22に順次に書き移す。次に縦仮線テーブル25に記憶されている縦仮線データを順次に読み出し、上記と同様に、始点のx座標の判定と、閾値 TH' との比較処理を行ない、有効データのみを擬線テーブル22に順次に追加する。この場合、擬線テーブル22に記憶された横線データの個数を変数 HNO 、縦線データの個数を VNO 、合計のデータ個数を $TOTAL$ としてカウントしておく。擬線テーブル22において、横線データのグループと縦線データのグループは、例えば始点と終点のx座標をチェックすることにより識別できる。始点と終点のx座標が同じ値であれば、縦線

特開平2-138674 (10)

データである。

第20図は、罫線テーブル22に記憶される始点データ22B、終点データ22Eと、実際の罫線上の位置との対応関係を示す。横罫線210Hの場合は、始点データ22Bと終点データ22Eは、最上位にある仮線の両端 P_A と P_B を指している。一方、縦罫線210Vの場合、これらのデータは、左側に位置する仮線の両端 $P_{A'}$ と $P_{B'}$ を指している。

第21Aと第21B図は、第3図に示す交点処理ルーチン310の詳細を示す。罫線テーブル22に得られた各罫線データは、第2E図に符号213'で示す如く、枠からはみ出した縦線、または横線のデータを含む交点処理ルーチンは、縦線と横線との交点部分における異常を検出し、はみ出した線の端点座標を修正するためのものであり、横線の修正処理(第21A図)と縦線の修正処理(第21B図)とからなる。

横線の修正処理では、横方向にはみ出した線分の端点x座標を、この横線を交差する縦線の位置

に一致させる。すなわち、各横線の始点、終点のy座標は変更せず、x座標だけを修正する。この処理は、罫線テーブル22から座標修正対象とする横線を順次に選択し、該横線の始点x座標に近いx座標をもつ縦線を探す。「近い」か否かの判断には、例えば、縦仮線と横仮線の識別の際に用いた閾値THを1つの基準値として用い、対象となる横線とx座標の距離がTH以内にある縦線が見つかったら、その縦線のx座標を上記横線の新しいx座標とする。終点についても同様に、近いx座標をもつ縦線を見つけ、横線のx座標を上記縦線のx座標で置き換える。この処理を全ての横線について繰り返すことにより、横罫線のはみ出し部分を除去することができる。

一方、縦線の修正処理では、x座標をそのままにして、y座標だけを変更する。すなわち、修正対象となる縦線を順次に選び、各縦線の始点と終点について、近いy座標をもつ横線を見つけ出し、縦線y座標を横線のy座標で置き換える。

第21A図は、上述の横線の修正の流れ図であ

る。iは横線用添字、jは縦線用添字、THはステップ503で求めた閾値である。ステップ1403では、ある横線に対して、その始点 $x_1(i)$ と近いx座標で定義される縦線を探している。

$$x_1(j) - L \leq x_1(i) \leq x_1(j) + L$$

上式の範囲内に始点 $x_1(i)$ があるときは、ステップ1404で、 $x_1(i) = x_1(j)$ と変更する。ステップ1405～1406は、終点について始点と同様の操作を行っている。ステップ1407は、始終点の変更に伴う長さの変更である。

第21B図は、前記縦線の修正処理の流れ図である。内容的には第21A図におけるxとyが入換わるだけで、アルゴリズムとしては同一である。すなわち、ステップ1423～1424は始点の変更、ステップ1425～1426は終点の変更、ステップ1427は長さの変更である。

次に、第3図の認識処理312の詳細について説明する。図は、例えば第22図に示す如く、縦横2本ずつの罫線210で囲まれたセル領域C1、C2、C3、C4……を意味し、各領域の大きさと位置は、各セル領域の左上画素(始点) Q_A と右下画素(終点) Q_B により定義される。本発明では、罫線の始点、終点と区別するために、図の始点 Q_A の座標を($cell_{-x_1}$, $cell_{-y_1}$)、終点 Q_B の座標を($cell_{-x_2}$, $cell_{-y_2}$)の如く表わすことにする。

第23図は認識処理と具体的に説明するための図であり、罫線210が、上から順に番号付けられた横線H1、H2……H7と、左から順に番号付けられた縦線V1、V2……V5からなる。認識処理312は、これらの罫線で左右、上下を囲まれたセル領域C1～C13を順次に認識し、各領域の始点と終点の座標を罫線テーブル23に記憶するものである。セル領域の認識は、まず最初の横線H1を基準線に設定して左側の縦線から順にチェックし、上記基準線と交じわる左辺、右辺

特開平2-138674 (11)

となるべき2本の縦線V1とV3を見つけ、次にこれらの縦線V1、V3と交じわり、上記基準線H1に最も近い下辺となるべき横線H2を探し出し、H1、V1、V3、H2で囲まれる領域を第1の欄(セル番号C1)と認識する。欄が認識されたら第22図で定義した始点Q_Aと終点Q_Bの座標を求め、これを欄テーブル23に登録する。次に、上記基準線H1を上辺、V3を左辺とするセル領域の右辺となるべき縦線V5を見つけ、縦線V3とV5に交じわる下辺となるべき横線H2を探し出して、H1、V3、V5、H2で囲まれた領域を第2の欄(セル番号C2)と認識する。認識されたセル領域の右辺が最も右側の縦線であった場合、上記基準線H1を上辺とする欄はこれ以上存在しない。そこで、第2番目の横線を基準線に設定して、上述したと同様のチェック動作を繰り返し、これによりセル領域C3、C4、……C11を認識する。基準線が最下位の横線となった時、上記動作は終了する。

第24図は、上記第23図の例におけるセル領

域の認識順序241と、各セル領域を定義する上辺242、左辺243、右辺244、下辺245に該当する罫線との関係を示す。

第25図は上記欄認識処理の具体的なプログラム・フローチャートを示す。このフローチャートでは、1本目の横線用にi、1本目の縦線用にj、2本目の横線用にkそして2本目の縦線用にlを、それぞれ、添字として用いている。また、説明の簡単化のため、罫線の線幅はすべてWであるとし、欄番号を示す添字をa、a番目の欄の左上点を($cell_x_1(a)$, $cell_y_1(a)$)、右下点を($cell_x_2(a)$, $cell_y_2(a)$)としている。

第25図において、ステップ1604～1605は、1本目の縦線、すなわち、欄の左辺候補を探すステップ、ステップ1607～1608は、2本目の縦線、すなわち、欄の右辺候補を探すステップ、また、ステップ1610は、2本目の横線、すなわち、欄の下辺候補を探すステップである。ステップ1611では、欄の左上点と右下点を定義し、欄テーブル23に座標デ

ータを記憶する。左上点のx座標は左辺のx座標+線幅で、y座標は上辺のy座標+線幅となる。また、右下点のx座標は右辺のx座標-1、y座標は下辺のy座標-1となる。

第26図に欄テーブル23の1例を示す。

23Nは欄(セル領域)の番号、23Bは始点座標、23Eは終点座標である。

以上説明した本発明の実施例によれば、文字と罫線とが混在する帳票を画像として入力し、罫線のみを自動的に抽出し、これをベクトルデータに変換している。また、各罫線により形成される欄の大きさと位置を自動的に認識し、欄定義データをテーブルに記憶している。従って、本発明によれば、画像入力装置において罫線が劣化した場合、あるいは帳票用紙自体が、例えば複写により劣化した罫線を含む場合でも、上記ベクトルデータにより鮮明な罫線を再生できる。また、各罫線がベクトルデータに変換されているため、通常の文書編集端末やワードプロセッサが備える文書編集機能を利用して、キーボードからのコマンド入力と

カーソル操作に应答して、任意の罫線の位置、長さ、線幅などを部分的に修正したり、新たな罫線の追加、既存罫線の削除などの加工を施すことができる。

本発明の変形例としては、例えば第3図の交点処理ステップ310あるいは欄認識処理ステップ312において、全ての罫線の線幅を所定の寸法に統一するようにしてもよい。線幅の決定方法としては、例えば、罫線テーブル22内の全ての罫線の線幅の頻度を求め、その中で最大頻度をもつ線幅に統一する方法、あるいは頻度の大きいn種の標準線幅を選択し、各罫線の線幅を最も近い標準線幅に一致させる方法など、種々の方法を採用できる。

また、第3図の文字外接矩形枠(文字枠)認識処理306で抽出した文字枠データを、罫線テーブル22と対応して用意した、文字枠テーブルに保存しておき、表示ステップ309で、あるいはステップ312の後で行なう表示ステップで、罫線と文字枠を重ねた形で表示するようにしてもよ

特開平2-138674 (12)

い。このようにすると、オペレータは、原入力画像中の文字位置と対応する位置に、カーソルを合せながら、キーボードから文字や記号を入力できる。

〔発明の効果〕

以上の説明から明らかな如く、本発明によれば、罫線以外の文字情報を含む原紙を用いて、画像処理により罫線のベクトルデータを抽出しているため、入力文書画像上の罫線が劣化していても鮮明な罫線を描くことができる。また、これら罫線をキーボード操作により修正したり、新たな罫線データを追加したりすることができるため、文書処理の効率を大幅に向上させることができる。

4. 図面の簡単な説明

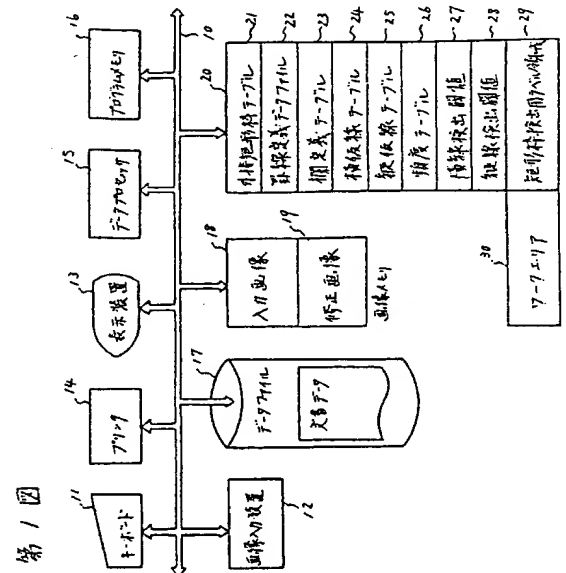
第1図は本発明による文書処理装置の全体構成を示す図、第24図は、本発明による文書処理における文書画像の変化の様子を示す図、第3図は本発明による文書処理を実行するためのプログラムのジェネラル・フローチャート、第4図は、第3図における傾き検出ルーチン302の詳細フロー

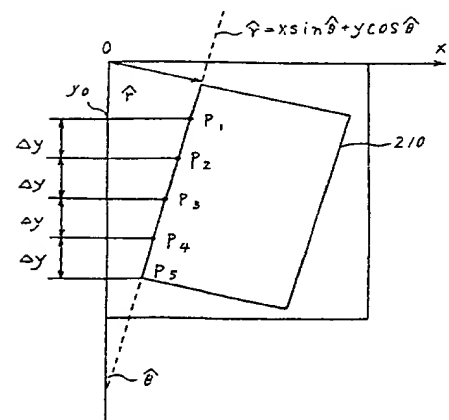
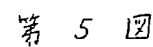
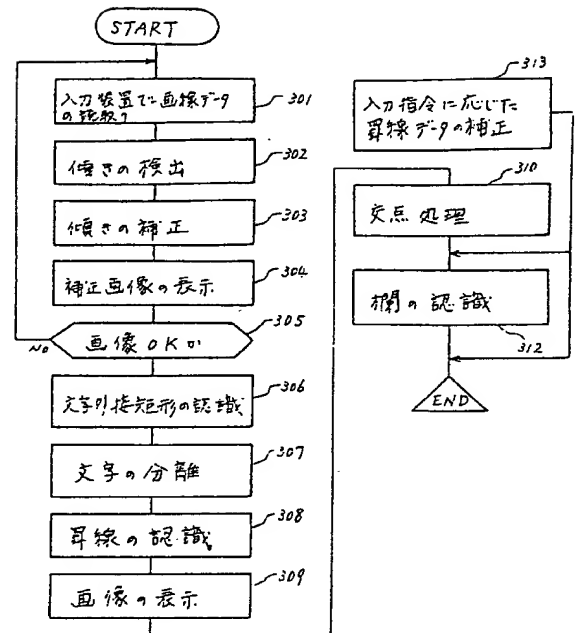
チャート、第5図は、文書画像の傾き検出を説明するための図、第6図は第3図の文書画像の傾き補正ルーチン303の説明図、第7A図と第7B図は、文書画像に含まれる文字領域の検出方法を説明するための図、第8図は文字外接矩形テーブルを示す図、第9A図と第9B図は、それぞれ、文書の原稿に含まれる罫線と、入力された文書画像に含まれる劣化した罫線とを示す図、第10図は第3図の罫線認識ルーチン308の詳細フローチャート、第11図は、第10図の罫線認識ルーチン500の詳細フローチャート、第12図は仮線定義データを記憶するためのテーブルを示す図、第13図は、仮線の長さとの関係を示す頻度分布図、第14図は第11図の仮線統合ルーチン504の詳細フローチャート、第15図と第16図はそれぞれ仮線統合のためのルールの定義を示す図、第17図は、仮線統合の手順を具体的に説明するための図、第18図は、仮線統合処理の途中における仮線テーブル24の状態を示す図、第19図は、複数の仮線を統合して得られる罫線

(実線)の定義データを記憶するテーブルの内容を示す図、第20図は罫線の定義データにおける始点と終点を示す図、第21A図と第21B図は、それぞれ第3図の交点処理ルーチン310を構成する傾線修正ルーチンと縦線修正ルーチンの詳細フローチャート、第22図は罫線の定義データにおける始点と終点を示す図、第23図は、表における罫線と罫との関係を説明するための図、第24図は罫の検出順序を示す図、第25図は、第3図における罫認識ルーチン312の詳細フローチャート、第26図は罫定義テーブル23の1例を示す図である。

符号の説明

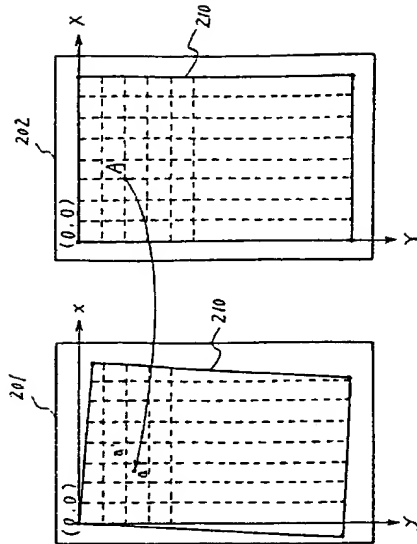
11…キーボード、12…画像入力装置、
13…表示装置、14…プリンタ、15…データプロセッサ、16…プログラム・メモリ、
17…データ・ファイル、18…入力画像メモリ、
19…修正画像メモリ、20…ワークメモリ、
210…入力画像上の罫線、215…描画された罫線。



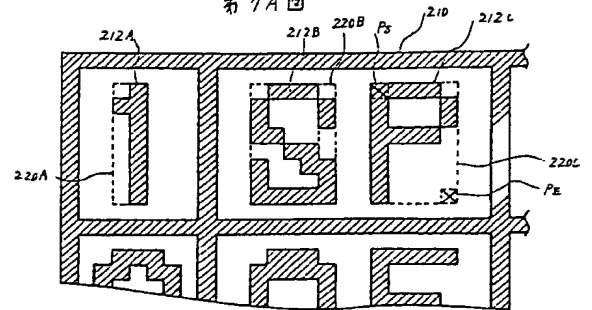


特開平2-138674 (14)

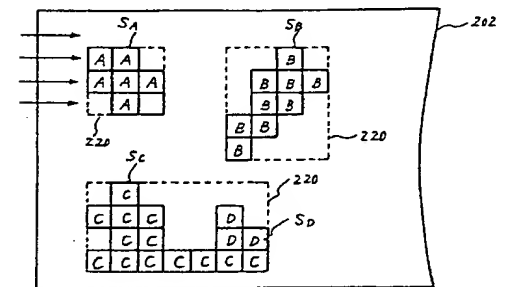
第6図



第7A図



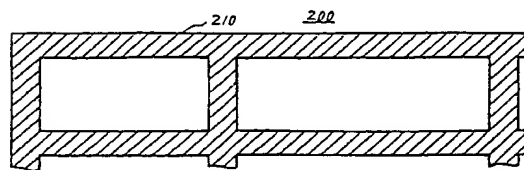
第7B図



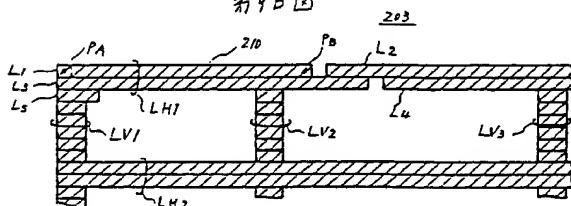
第8図

矩形番号	座標 (X, Y)	
	始点	終点
1	(5, 10)	(10, 20)
2	(60, 65)	(80, 80)
...

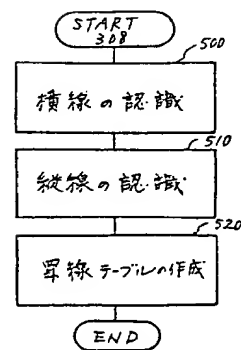
第9A図



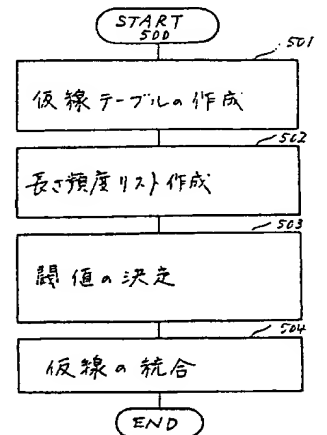
第9B図



第10図



第11図



特開平2-138674 (15)

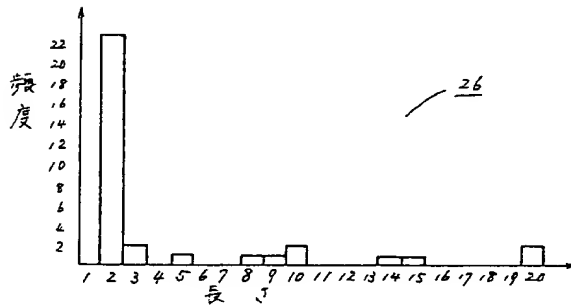
第12図

仮線テーブル

線番号	座標 (x, y)		線幅	線長
	始点	終点		
1	(1, 1)	(4, 1)	1	3
2	(1, 2)	(4, 2)	1	3
3	(1, 3)	(5, 3)	1	4
4	(10, 4)	(14, 4)	1	4
⋮	⋮	⋮	⋮	⋮

24N 24B 24E 24W 24L

第13図



第15図

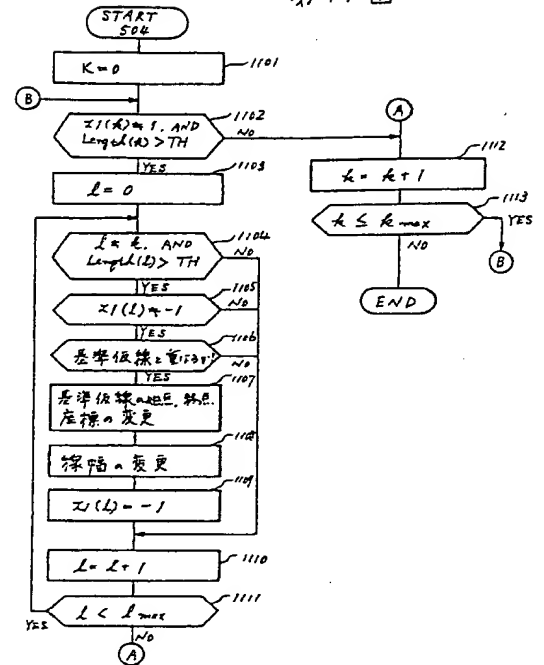
CASE No	重複関係	端点座標の定義
1	$x_1(A) \dots x_2(A)$ $x_1(L) \dots x_2(L)$	$x_1(A) = x_1(L)$ $x_2(A) = x_2(L)$
2	$x_1(A) \dots x_2(A)$ $x_1(L) \dots x_2(L)$	$x_1(A) = x_1(L)$ $x_2(A) = x_2(L)$
3	$x_1(A) \dots x_2(A)$ $x_1(L) \dots x_2(L)$	$x_1(A) = x_1(L)$ $x_2(A) = x_2(L)$
4	$x_1(A) \dots x_2(A)$ $x_1(L) \dots x_2(L)$	$x_1(A) = x_1(L)$ $x_2(A) = x_2(L)$

第16図

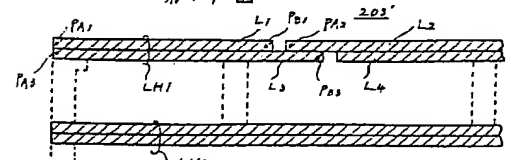
CASE No	重複関係	線幅の定義
1	$w(A) \dots w(L)$	$w(A) = w(L)$
2	$w(A) \dots w(L)$	$w(A) = w(L) + 1$
3	$w(A) \dots w(L)$	$w(A) = w(L)$

W: width

第14図



第17図



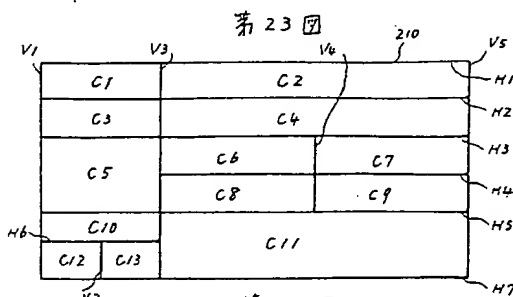
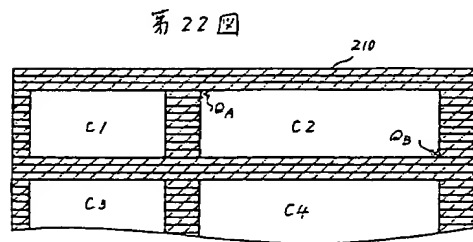
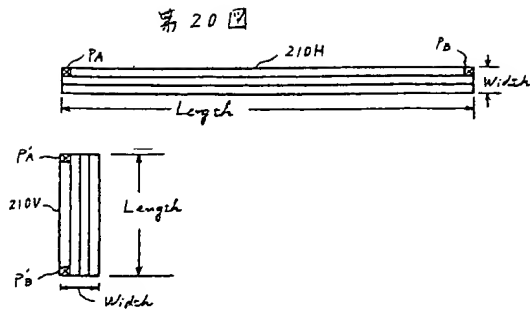
第18図

CASE No	座標 (x, y)	線幅	線長
1	(1, 1)	3	4
2	(1, 2)	1	3
3	(1, 3)	1	4
4	(10, 4)	1	4
⋮	⋮	⋮	⋮

第19図

CASE No	座標 (x, y)	線幅	線長
1	(1, 1)	3	4
2	(10, 4)	1	4
⋮	⋮	⋮	⋮

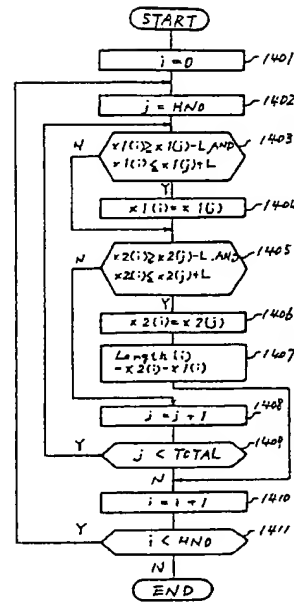
22B 22E 22W 22L



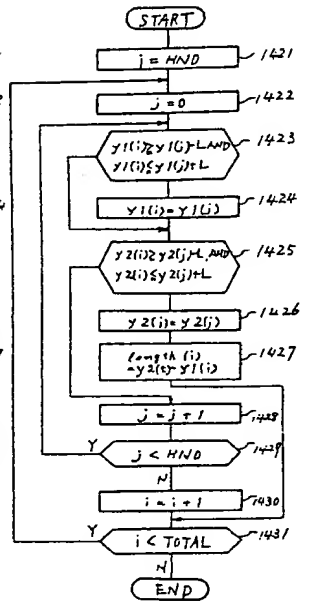
第24図

	上辺	左辺	右辺	下辺
C1	H1	V1	V3	H2
C2	H1	V3	V5	H2
C3	H2	V1	V3	H3
C4	H2	V3	V5	H3
C5	H3	V1	V3	H5
C6	H3	V3	V4	H4
C7	H3	V4	V5	H4
C8	H4	V3	V4	H5
C9	H4	V4	V5	H5
C10	H5	V1	V3	H6
C11	H5	V3	V5	H7
C12	H6	V1	V2	H7
C13	H6	V2	V3	H7

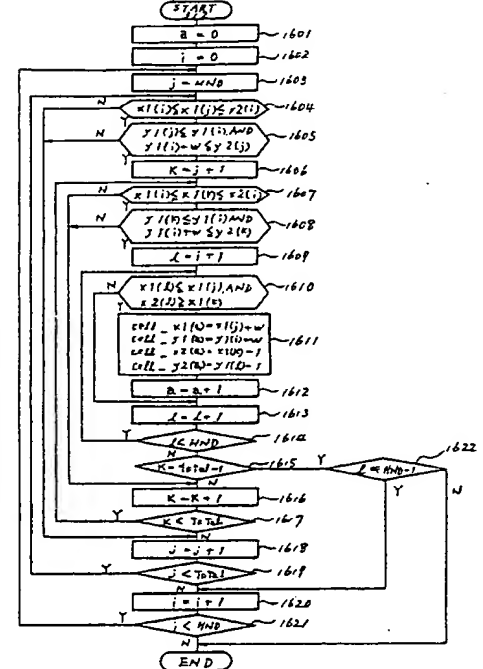
第21A図



第21B図



第25図



特開平2-138674 (17)

第26図

23

セルNo	座 標	
	始 点 (Cell-x1, Cell-y1)	終 点 (Cell-x2, Cell-y2)
0	(1, 1)	(15, 8)
1	(1, 10)	(15, 18)
2	(1, 20)	(15, 28)
3	(1, 30)	(15, 38)
⋮	⋮	⋮

23N
23B
23E

Part translation of JP

Published Japanese Patent Application No: H02-138674

(Lines from line 13 in top left column to line 1 in top right column on page 4)

According to the present invention, the extraction of horizontal ruled lines is progressed, for example, in a manner in which line segments shorter than a threshold length among the line segments each consisting of more than one black pixels detected by scanning the second document image in a horizontal direction are exempted from the extraction. And the extraction of vertical ruled lines is progressed, for example, in a manner in which line segments shorter than a threshold length among the line segments each consisting of more than one black pixels detected by scanning the second document image in a vertical direction are exempted from the extraction.

=End=